Synthetic Training Data for Visual Model Enhancement

Igor Overchuk

CSCI 5922, University of Colorado Boulder

Abstract. Lack of training data is a persistent issue across all sectors of machine learning. It has been shown that having more data to train with will almost always improve model performance, regardless of modality or model size. In the visual domain, generative models have seen a surge in performance in the last couple of years, and even small diffusion models that can run on your laptop can generate realistic images given a text prompt. The question this study aims to answer is, can generative models produce images good enough to be used as training data and enhance the performance of visual models? To test this out, I'll be using an animal dataset with 90 classes and 60 images per animal class, a challenging and scarce domain. The real images will be split 30/30 into train/test sets. A stable diffusion model will be used to generate 30 synthetic images per animal class. These will only be used as training data. Two models are trained, a simple CNN, and a fine tuned pre-trained model, ResNet50. The training dataset is split into 5 different configurations to test how different proportions of real to synthetic data perform. The results are promising, as the basic CNN model improved 1.5x on average across all metrics when using 30 real and 30 synthetic images during training as compared to using only the real images. The ResNet50 model did not see such improvements, improving by about 0.5-1.0% on average. However, it did see a lower test loss, suggesting that the model was able to classify with a higher confidence. Another key observation is that when synthetic images outnumbered real ones, the performance of the models fell. This means it is crucial to have an even or higher ratio of real images if synthetic data is to be introduced into the training process. Overall, inserting synthetic data into the training process of visual models has a positive impact on performance, leading to higher accuracy and lower loss, helping the models better generalize with the extra variability during training.

1 Introduction

Currently, there are a variety of machine learning algorithms in use. They differ in architecture, input and output, size, and goal, but all of them need one thing to be effective. Training data. As hardware gets faster, models get larger, and scaling laws prove that models trained with more data perform better, the search for good, original data becomes a growing issue in this space. In my own personal experience during my undergraduate capstone project, our custom image classification model failed due to a lack of training data in a very specific domain, forcing us to find another solution. In recent years, generative models have become the focus of the AI world. They offer surprisingly accurate material, whether that be text, image, or a suite of other modalities. The question this experiment aims to answer is the following: Can AI-generated images used as training data enhance the performance of visual classification models? Can the vast knowledge of large generative models be used to tune a smaller model designed for a specific task?

One solution for adding training data is data augmentation. Flipping images, cropping, or slightly altering lighting can create enough variance in a single training sample to create several. This is a great technique for dealing with scarce image data, but it has limits. There are only so many new samples that can be created before the benefits level out and the model reaches its performance ceiling given that training set. To extend beyond that, original training data is needed, and generative image AI looks like a promising source. Another option to get great results with limited training data is Few-Shot Learning with LLMs. A user can simply prompt the LLM with the training data and its labels followed by a classification question and get impressive results. This is a great concept, but an inefficient solution, as prompts to LLMs are extremely computationally expensive. Finding a way to enhance a smaller model could be crucial for applications that require a lower processing demand. Notably, there is already research in the area of synthetic training data that looks very promising, but due to the recency of realistic generative models, it is a very active area of study.

The proposal is that it is possible to enhance the performance of compact visual models using synthetic image data. This is a one-time investment in expensive compute on calls to a very large model that will enable a simpler model to be put out to end users, which will reduce the computing resources used in the long run. As there is previous research on this, my experiment aims to offer a greater understanding and additional proof of concept for this idea. As generative AI can create limitless original images in various domains, this method should exceed the performance gains of data augmentation, which has a finite limit. And, as this will train a smaller fine-tuned model, it will exceed the energy efficiency of directly calling LLMs.

2 Related Work

The main topic of this study is *Using Generative AI to Create Synthetic Training Data to Enhance Models*. This topic explores various methods/use cases of training visual models with synthetic data. [1] **This Dataset Does Not Exist: Training Models from Generated Images** looks at training on synthetic data alone with promising results. My study aims to test various combinations of training subsets to see which can be the most effective. [2] **AI-Generated Images as Data Source: The Dawn of Synthetic Era** explores generating images for edge case models without a lot of real world data. This is similar to what is being studied here, but my focus isn't on a specific edge case. [3] **A novel algorithm can generate data to train machine learning models in conditions of extreme scarcity of real world data** explores the design of an additional genetic algorithm to improve the output of generative models in cases of extreme scarcity. In this study, breast cancer detection. My study does not aim to improve synthetic image generation, rather it looks to use existing generative models as they are.

Another related topic is *Model Degradation When Training on AI Generated Data*. This research warns us of the inevitable mixing of synthetic and real-world data out on the internet, where most large models get their data from right now. [4] **AI models collapse when trained on recursively generated data** and [5] **On the caveats of AI autophagy** study the performance of these models as more and more synthetic data is injected into the training process and find that eventually the ratio of real/synthetic data becomes too great, and the model collapses. They focus mainly on LLMs, so my study focusing on image models differs, but I'll be monitoring for potential degradation as the percentage of synthetic data increases as well.

Finally, it is also important to consider the *Ethical Implications of Using Synthetic Data*. This research focuses on the ethical and political implications of training models with synthetic data. [6] **The Use of Synthetic Data to Train AI Models: Opportunities and Risks for Sustainable Development** and [7] **Machine learning and the politics of synthetic data** study how biases from the GANs used to supply the generated data can propagate into the models that the data is used to train. Also, they talk about the issues of synthetic data ownership when using generative models. This study does not dive into these ethical considerations in depth, but it is very important to consider these points as they can have significant implications as society becomes more and more integrated with these algorithms.

3 Methodology

I used a simple domain that has limited training data to simulate a scarce data environment with a manageable task. The dataset is from Kaggle. It has 90 different animal categories, with 60 images total per animal. That makes 5400 images in the entire dataset, small by today's standards. I split the dataset into 30 training images and 30 test images per animal class to create a fairly scarce training environment, as 30 images per category is unlikely to be enough to achieve great results with small models.

To generate the synthetic images, I used ComfyUI, a software application that is installed locally on your machine and uses your graphics card to run a generative model of your choice to create images from prompts. The generative model I am using is Stable Diffusion 1.5. Using my laptop NVIDIA RTX 2060 GPU, it takes about 5 seconds to generate one 512x512 image. I have used this to generate 30 synthetic images per category to achieve a 50/50 real/synthetic split in my training set. Alternatively, I explored using state of the art models, such as ChatGPT's image generation and Midjourney, but their daily limits would make it difficult to create a meaningful amount of additional images in the time we had to perform the experiment. Figure 1 shows how some of these synthetic images look.

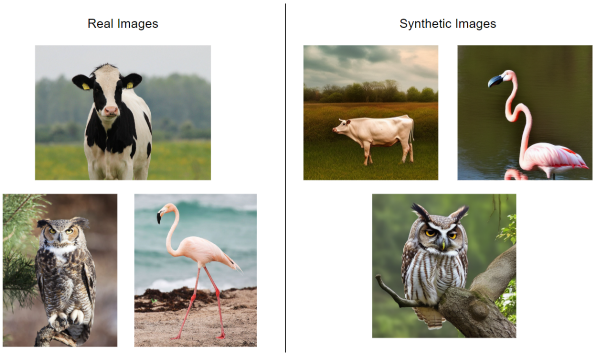

Fig. 1. Comparison of real training data to synthetic training data

Comparing 3 of the animal classes, we can see that synthetic data is fairly accurate, but contains mistakes. Usually, these mistakes are the shape of the animal, placement of limbs, structure of the face and so on. However, not all generated images have mistakes, and this points to a need for a manual vetting process, where synthetic data is reviewed before it is used in training. I did not go through such a process in order to see how these mistakes would affect training. The prompt I used to generate the images was the same for each animal, and is as follows: "natural scenery, wildlife photo, image of a ¡animal¿".

The models I used to test performance are a simple CNN, to see how a really small model reacts to the extra data, and a combined pre-trained model using ResNet50, to see how a larger capacity model reacts. I did not have time to test with a visual transformer, which was listed as an optional third model on the proposal.

As seen in Figure 1, the CNN model has 5 convolutional/max pool layers, each adding 8 channels and decreasing the frame by a factor of 2. After the 5th layer, there are 2 fully connected layers, the CNN output is flattened to a size of 640, then connected to a layer of 128, and then the second fc layer down to the number of classes 90. The total number of trainable params is 117,034. For regularization, there is an L2 penalty with weight decay at 0.0001 and dropout is applied to all layers, with a probability of 0.1 for convolutional and 0.5 for fully connected. The learning rate is 0.0075 with the Adam optimizer, and the model is trained for 50 epochs, as anything past this leads to overtraining. Cross entropy loss is used as the loss metric.

As seen in Figure 2, ResNet50 is a fairly large and complex network. In this study, the loaded pre-trained weights that it got from training on millions of

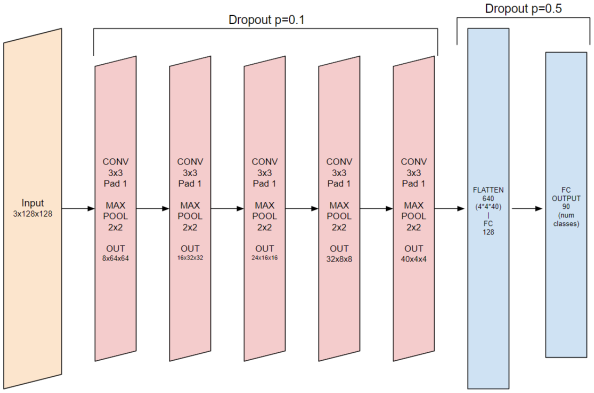

Fig. 2. Architecture of the simple CNN

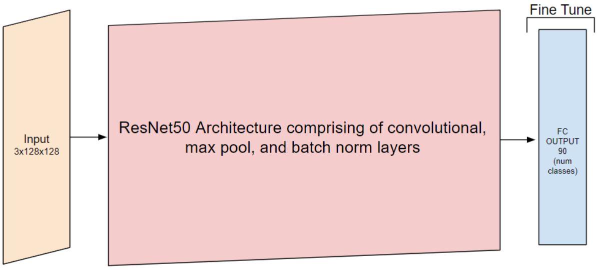
Fine Tune
Input
3x128x128
ResNet50 Architecture comprising of convolutional, max pool, and batch norm layers
FC
OUTPUT
90
(num
classes)

Fig. 3. Architecture of fine tuned ResNet50

images from ImageNet will be used. The only modification that was made was to the final fully connected layer. By default, it has 1000 output classes, but has been modified to instead have 90 output classes. The pre-trained weights of the model are frozen, and thus the only trainable params are in the final fully connected layer, with a total of 184,410 params. For regularization, there is an L2 penalty with weight decay at 0.001. The learning rate is 0.001 with the Adam optimizer, and the model is trained for 20 epochs, as it converges quite quickly and additional epochs aren't necessary. Cross entropy loss is used as the loss metric.

4 Experiments

This study performed a single experiment to see how various amounts of synthetic training data affect performance on a test set of real images. The real data is split into 30 training and 30 testing images. The models were hyper parameter tuned using real data only, and once the optimal parameters were found, they were kept constant for all other training runs. There were 5 different training set configurations, with varying amounts of real/synthetic images. The 5 configurations were:

- 30 real and 0 synthetic
- 30 real and 15 synthetic
- 30 real and 30 synthetic
- 15 real and 30 synthetic
- 0 real and 30 synthetic

The metrics used to evaluate each model run are the basic accuracy, precision, recall, and F1 score.

First, taking a look at the accuracy and loss curves of both models with the 30 real 0 synthetic vs 30 real 30 synthetic training configurations shows promising results.

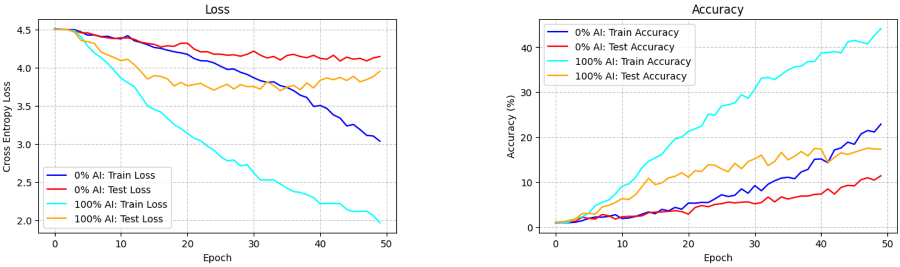

Fig. 4. Loss and Accuracy Curves of the CNN

For the CNN, in figure 4, a significant improvement can be seen in both accuracy and loss, with the synthetic data model converging quicker and to a better accuracy/loss.

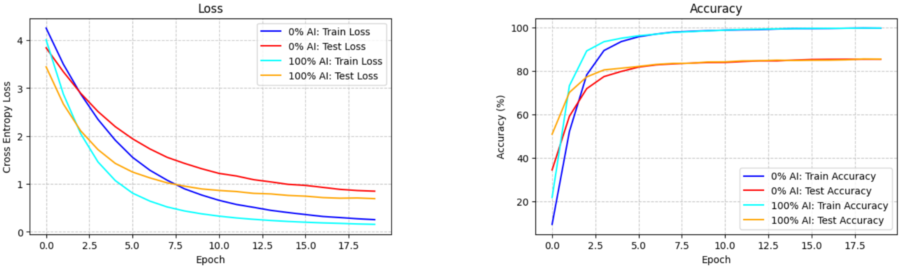

Fig. 5. Loss and Accuracy Curves of ResNet50

For ResNet50, in figure 5, a much smoother curve is formed, with the synthetic data model improving in the loss metric, suggesting it is making its predictions with a higher confidence. Unfortunately, the accuracy of the two models stayed around the same, with both reaching near perfect training set accuracy and maxing out around 85% for the test set. I believe the limited performance in test accuracy is due to the fact that both training configurations reach almost 100% training accuracy, and generalize similarly to the test set, as the model structure is the same.

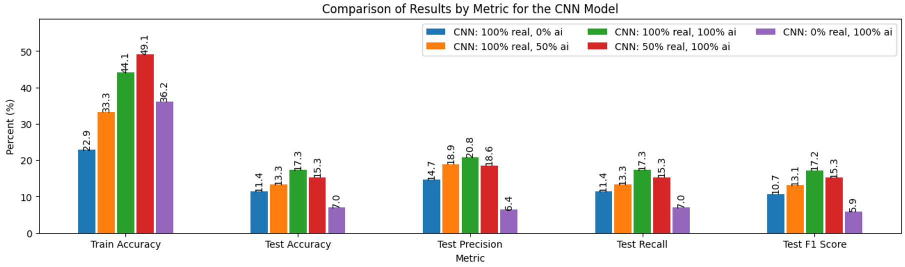

Fig. 6. Model Metrics for the CNN

In figure 6, looking at the the performance of all 5 training configurations for the CNN, a clear trend can be seen where all of the test metrics improve by about 1.5x when training with the full synthetic data. Another important trend to note is the dropoff in performance when training on synthetic data alone, suggesting that the quality of the synthetic data is lower than real.

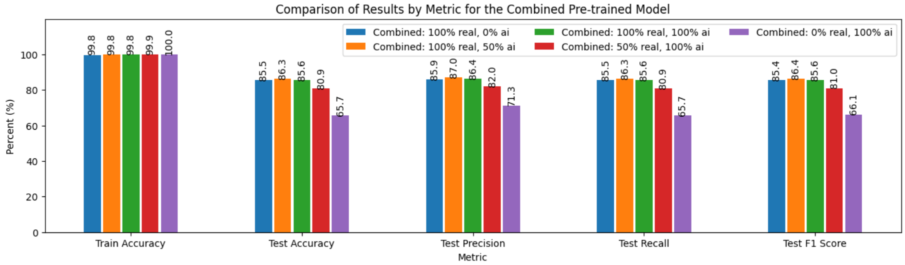

Fig. 7. Model Metrics for ResNet50

In figure 7, looking at the performance metrics of the ResNet model, it can be seen that each metric is improved by roughly 0.5-1.0%. The interesting thing about these results is that the best performing model is the one with 30 real and 15 synthetic training images, suggesting that when using synthetic data, it is important to keep the amount of real data higher to get the best results. Similarly to the CNN model, a performance drop can be seen when there is more synthetic data than real, reinforcing the need for that balance.

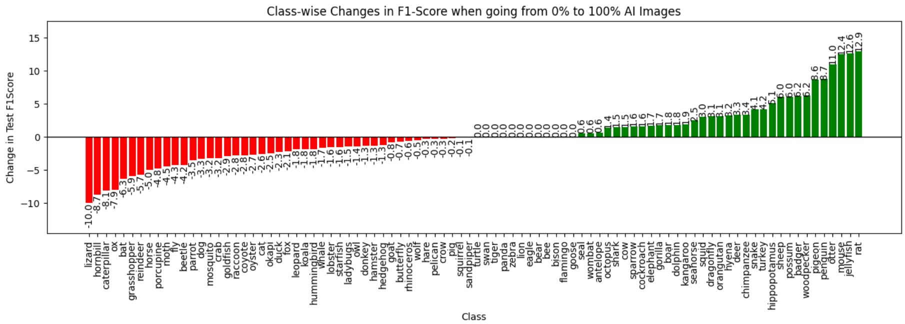

Fig. 8. Classwise changes in F1 Score for ResNet50

Finally, in figure 8, let's look at class by class performance differences on the ResNet50 model to see if there are trends among classes that improved and those that didn't. The red indicates a drop in F1 score when going from 0 synthetic to 30 synthetic images, and the green an improvement. It's interesting that the ratio of classes that improved and those that didn't is fairly even. I was expecting to see minor improvements across all classes, with a few outliers that got worse, but that is not the case.

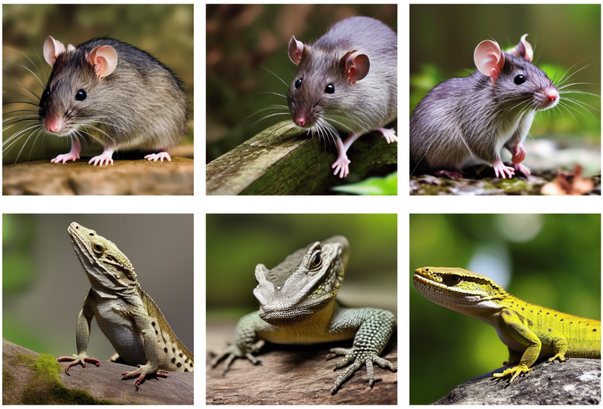

Fig. 9. Generated images of rats and lizards

In figure 9, looking at generated images for the most improved class, rat, and the class that dropped in performance the most, the lizard, it is hard to find significant differences in the quality of these images. The rat pictures do look very realistic and do not contain any major mistakes. However, the lizard images also look fairly realistic, with the exception of the middle image. It is hard to definitively say that the quality of these images is what caused their classes to perform better/worse.

There are several additional points of research that could be done around this study. First would be hyper parameter tuning with each set of training data individually as opposed to keeping model params constant in order to extract the most out of each configuration. Another question is how the model would perform if each generated image was manually vetted and removed if it contained mistakes or did not look realistic. An interesting study would be to compare a set of synthetic images that are unvetted to ones that are and see if there is a performance difference. Finally, the generative model used for this experiment was fairly small and all images were generated using the same model. What if the synthetic images were generated using several state of the art models? Would having more than one generative model produce better variability?

5 Conclusions

Synthetic data is a viable source of original training data in the visual domain. It has the ability to significantly improve simple models trained from scratch, and enhance the performance of pre-trained models as well. For visual domains that lack training data, it can be used alongside well known strategies like data augmentation to enlarge a scarce training dataset. Even without vetting the synthetic data and using a small generative model, the performance gains were impressive. On the flip side, while synthetic data has proven to be very useful, it has also shown it is not as good as real data and should be treated as an inferior, auxiliary source. This means that the ratio of real/synthetic images must be at least even, with real images ideally outnumbering synthetic. If the model sees too much fake data during training, it will start to prioritize the features of the generative model, and will not generalize well to the real test set.

The main ethical implications of using synthetic data is those of inaccuracy. When training on real data, it's guaranteed that the training samples are correct and have occurred and been captured in the real world. With synthetic data, there is no such guarantee. So for example in a medical field, a model making an incorrect determination due to a synthetic training sample could have severe implications. It is hard to tell which training sample most influenced the model on a particular call, so who's to be held accountable for the model's decisions if it is incorrect? Another ethical consideration is data ownership. Synthetic visual data offers an original source of data that does not have a clear owner, other than maybe the person that prompted the generative model. However, there is a question of whether the owners of the images used to train the generative model should have an ownership stake in the images that are generated. This is a complex question with no easy answer. In terms of fairness, it is known that large language models are prone to bias, as they are trained on largely unfiltered human generated data. These biases can propagate to the data they generate, and if that is then used to train other models, we may have unwanted amplification of these biases. Finally, there is a question of computational costs when calling LLMs, as each call consumes a ton of energy. If synthetic data is introduced as a training source, there would be another wave of energy usage just to generate data to train models on, adding to the already high energy usage of the AI industry today.

References

1. Besnier, V., Jain, H., Bursuc, A., Cord, M., Pérez, P.: This dataset does not exist: training models from generated images. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE (2020) 1–5
2. Yang, Z., Zhan, F., Liu, K., Xu, M., Lu, S.: Ai-generated images as data source: The dawn of synthetic era. arXiv preprint arXiv:2310.01830 (2023)
3. Niel, O.: A novel algorithm can generate data to train machine learning models in conditions of extreme scarcity of real world data. arXiv preprint arXiv:2305.00987 (2023)
4. Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R., Gal, Y.: Ai models collapse when trained on recursively generated data. Nature **631**(8022) (2024) 755–759
5. Xing, X., Shi, F., Huang, J., Wu, Y., Nan, Y., Zhang, S., Fang, Y., Roberts, M., Schönlieb, C.B., Del Ser, J., et al.: On the caveats of ai autophagy. Nature Machine Intelligence (2025) 1–9
6. Marwala, T., Fournier-Tombs, E., Stinckwich, S.: The use of synthetic data to train ai models: Opportunities and risks for sustainable development. arXiv preprint arXiv:2309.00652 (2023)
7. Jacobsen, B.N.: Machine learning and the politics of synthetic data. Big Data & Society **10**(1) (2023) 20539517221145372